

Hanbit eBook

Realtime 51



Xcode 5, 플랫 UI, API 중심으로 살펴보는 iOS 7

iOS 7 핵심 노트

아곰 지음

Xcode 5, 플랫 UI, API 중심으로 살펴보는 iOS 7

iOS 7 핵심 노트

iOS 7 핵심 노트 Xcode 5, 플랫 UI, API 중심으로 살펴보는 iOS 7

초판발행 2013년 12월 31일

지은이 야곰 / **펴낸이** 김태현

펴낸곳 한빛미디어(주) / 주소 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부

전화 02-325-5544 / **팩스** 02-336-7124

등록 1999년 6월 24일 제10-1779호

ISBN 978-89-6848-657-9 15000 / **정가** 11,000원

책임편집 배용석 / **기획** 이종민 / **편집** 김연숙

디자인 표지 여동일, 내지 스튜디오 [임], 조판 박진희

마케팅 김형진, 김진불, 조유미 / **마케팅** 박상용, 서은옥, 김옥현

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.

한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2013 HANBIT Media, Inc.

이 책의 저작권은 야곰과 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다. **지금 하지 않으면 할 수 없는 일이 있습니다.**

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

지은이_ **야곰**

yagom's blog(<http://blog.yagom.net/>)를 운영하는 iOS 개발 파워 블로거이자 iOS 프로그래머. 아이폰 개발자 커뮤니티인 맥부기(<http://cafe.naver.com/mcbugi>)에서 강좌를 연재 중이다. 컴퓨터 교육을 전공했으며 비전공자와 학생들에게 컴퓨터 지식을 더 쉽고 재미있게 알리는 데 관심이 많다. 2010년부터 iOS 개발을 시작해 현재까지도 계속 iOS 관련 개발에 열정을 쏟고 있다. 내일 걱정은 모레하는 것이 좋다고 생각하며 스스로 긍정적인 마음가짐을 빼면 시체라고 말한다. 스스로 개발자라고 생각하지 않는 것을 보면 괴짜임이 틀림없다. 무엇보다 여행과 요리를 좋아한다.

저자 서문

CEO가 바뀌고 철학이 바뀌면서 주춤할 것 같았던 애플이 예상 밖으로 전진 행보를 계속하고 있습니다. iOS 7으로 업데이트 후에 큰 반발이 있을 것이라는 저의 예상과는 달리, 많은 사람들이 iOS 7을 반기고 있습니다.

제가 처음 iOS 개발을 할 때만 해도 iOS 개발 입문자가 많았었는데, 우리나라의 경우 iOS 점유율이 높지 않기 때문인지 iOS 개발 입문자들이 점점 줄어드는 것 같아 안타깝게 생각합니다. 따라서 이 책이 iOS 7 입문자와 중급자 분들께 조금이나마 도움이 되었으면 하는 바람입니다.

그리고 서문을 빌어 처음 한빛미디어와 좋은 인연을 맺을 수 있도록 길을 열어준 조희진 과장님과 이 책을 세상에 선보일 수 있도록 끝까지 많은 도움을 준 한빛미디어의 이종민 대리님께 감사의 말을 전합니다.

항상 곁에서 응원해주고 어떠한 도움도 아끼지 않는 가족, 동료 여러분들께 사랑한다는 말을 전하고 싶습니다. 앞으로도 함께 행복하게 살아갈 수 있기를 진심으로 소망합니다.

집필을 마치며

아곰

대상 독자 및 참고 사항

초급

초중급

중급

중고급

고급

이 책은 iOS 7의 핵심 개념을 소개하는 책입니다. iOS 개발의 기본을 알고 있으며 iOS 7에 관심이 있는 분이라면 누구나 읽을 수 있습니다. 또한 이 책의 샘플 코드를 실행하려면 다음에 소개하는 환경이 갖춰져 있어야 합니다.

- Mac OS X 매버릭스가 실행되는 매킨토시 컴퓨터
- Xcode 5를 사용할 수 있는 개발 환경

4장에서 소개하는 예제의 소스 코드는 <http://www.hanbit.co.kr/exam/2657>에서 다운로드할 수 있습니다.

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. eBook Only - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내시는 선배, 전문가, 고수분에게는 보다 쉽게 집필하실 기회가 되리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. 무료로 업데이트되는, 전자책 전용 서비스입니다.

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정한 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나, 저자(역자)와 독자가 소통하면서 보완되고 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여, DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT기기에서 자유롭게 활용하실 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해, 독자 여러분이 언제 어디서 어떤 기기를 사용하시더라도 편리하게 전자책을 보실 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 계실 가능성이 높습니다. 이 경우 저작권법에 의해 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한 한빛미디어 사이트에서 구입하신 후에는 횡수에 관계없이 다운받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구입하신 독자에게만 제공되며, 다른 사람에게 양도나 이전되지 않습니다.

차례

01	iOS 7	1
<hr/>		
	1.1 Xcode 5.....	1
	1.2 플랫폼 UI.....	2
	1.3 64비트 지원.....	5
02	Xcode 5	7
<hr/>		
	2.1 새롭게 추가된 템플릿.....	7
	2.2 새로워진 도큐먼트.....	8
	2.3 Source Control.....	10
	2.4 편리해진 계정 관리.....	11
	2.5 Capabilities.....	17
	2.6 Open Quickly.....	18
	2.7 옵션 설정이 쉬워진 검색 내비게이터.....	20
	2.8 인터페이스 빌더.....	21
	2.9 애셋 카탈로그.....	24
	2.10 사용자화 도큐먼트.....	25
	2.11 더욱 강력해진 컴파일러와 디버거.....	26
	2.11.1 모듈.....	26
	2.11.2 자동 백터화.....	27
	2.11.3 데이터 팁.....	28
	2.12 발전하는 디버깅.....	29

3.1	플랫 UI의 콘셉트.....	33
3.2	달라진 기본 컴포넌트.....	33
3.2.1	상태 바.....	34
3.2.2	내비게이션 바.....	34
3.2.3	검색 바.....	35
3.2.4	탭 바.....	35
3.2.5	툴 바.....	36
3.2.6	바 버튼.....	36
3.2.7	스크롤 뷰.....	37
3.2.8	스플릿 뷰 컨트롤러.....	37
3.2.9	테이블 뷰.....	38
3.2.10	피커 뷰.....	38
3.2.11	기본 버튼 모양들.....	39
3.2.12	페이지 컨트롤.....	39
3.2.13	프로그레스 뷰.....	39
3.2.14	리프레쉬 컨트롤.....	40
3.2.15	라운드-렉탱글 버튼.....	40
3.2.16	슬라이더.....	40
3.2.17	스테퍼.....	41
3.2.18	스위치.....	41
3.2.19	텍스트 필드.....	41
3.2.20	액션 시트.....	42
3.2.21	얼럿 뷰.....	42
3.3	iOS 7용 아이콘.....	43

4.1 Air Drop.....	45
4.1.1 Air Drop 예제.....	45
4.2 다이내믹스.....	51
4.2.1 주요 클래스.....	51
4.2.2 다이내믹스 예제.....	52
4.3 텍스트 키트.....	59
4.3.1 텍스트 키트 예제.....	60
4.4 백그라운드 페치.....	66
4.4.1 주요 클래스.....	66
4.4.2 백그라운드 페치 예제.....	66
4.5 P2P 연결.....	70
4.5.1 주요 클래스.....	71
4.5.2 P2P 연결 예제.....	73
4.6 사파리 서비스 프레임워크.....	84
4.6.1 주요 메서드 소개.....	84
4.6.2 사파리 서비스 예제.....	84
4.7 스프라이트키트 프레임워크.....	89
4.7.1 스프라이트키트 프레임워크의 구조.....	90
4.7.2 프로젝트 템플릿 구성.....	92
4.7.3 주요 클래스.....	93
4.7.4 움직임 및 물리 효과를 제어하는 클래스.....	96
4.7.5 라이프 사이클.....	101
4.7.6 스프라이트키트 예제.....	102

5.1 64비트 앱의 포인터.....	118
5.2 시스템 프레임워크의 분리.....	118
5.3 Xcode 5의 64비트 지원.....	119
5.4 64비트 전환 방법.....	119
5.4.1 데이터 타입의 변화가 앱에 끼칠 수 있는 영향.....	120
5.4.2 ILP32와 LP64 변환.....	120
5.4.3 데이터 타입 변경 시 주의 사항.....	121
5.4.4 그 외 주의사항.....	121

6.1 Exchanges 도입.....	124
6.2 리더보드 수의 제한 증가.....	124
6.3 하위 도전 과제 항목 추가 가능.....	124
6.4 부정 행위 방지 기능 추가.....	124
6.5 변경된 API.....	124

1 | iOS 7

약 1년 주기로 업데이트되는 iOS, 2013년 하반기에도 어김없이 새로운 버전의 iOS가 세상에 빛을 드러냈습니다. 지난 iOS의 어떤 업데이트보다 파격적이라는 생각이 들 정도입니다.

디자인 측면에서 살펴보면 플랫 UI라는 새로운 디자인 개념이 도입되었고 많은 기능 부분이 추가/변경되었습니다. 또한 iOS 업데이트와 함께 Xcode도 업데이트되었고, OS X도 업데이트되었습니다. 이런 대대적인 변화 속에 개발자들은 새로운 환경이 반가우면서도 두렵기만 합니다. 항상 그렇듯 매번 새로운 속제에 당면하게 되는 것이지요.

이 책은 여러분과 제가 당면한 이 새로운 과제를 헤쳐나가기 위해 준비했습니다. 우리는 새로워진 개발 환경을 시작으로 외형적으로 바뀐 iOS 7, 그리고 새로운 기능을 추가한 iOS 7을 차례대로 만나볼 것입니다. 이 책에서 다루는 내용은 너무 넘치지도, 너무 부족하지도 않은 내용이 될 것입니다.

1장은 iOS 7의 핵심 변경 사항에 관한 필자의 간략한 의견을 담았습니다. 가벼운 마음으로 개발자 입장에서 iOS 7의 무엇에 주목해야 하는지를 알아두면 좋을 것으로 생각합니다.

1.1 Xcode 5

2011년 초 새롭게 버전업 했던 Xcode가 약 2년 6개월만에 버전 5를 선보였습니다. Xcode 3에서 Xcode 4로 업데이트했다는 사실은 iOS 개발자 입장에서는 굉장히 큰 변화였습니다. Xcode 4는 맥 개발 센터에서만 지속적인 베타 테스트를 진행했었고, iOS에서는 계속 Xcode 3 위주로 베타 버전을 공개했습니다. 그런데

iOS 4.2를 처음 정식으로 공개하면서 그간 iOS 개발 센터에서는 테스트하지 않았던 Xcode 4를 전격적으로 공개했기 때문입니다. 기본적인 인터페이스부터 많은 변화가 있었지만 이러한 갑작스러운 공개도 혼란의 원인이 되었습니다. 하지만 Xcode 5는 Xcode 4를 처음 공개했을 때만큼 심한 변화는 아니며 넓은 관점에서 보았을 때 11가지의 대표적인 기능 변경 및 추가가 있었습니다.

자세한 사항은 '2장. Xcode 5'에서 살펴봅니다.

그림 1-1 Xcode 5

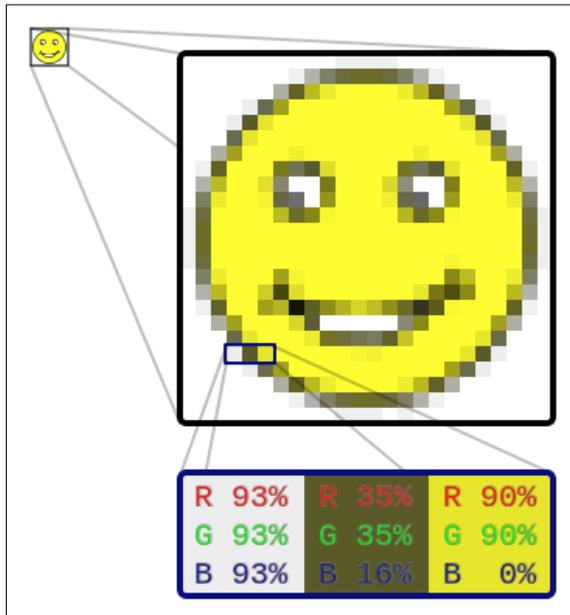


1.2 플랫폼 UI

플랫폼 UI는 이번 iOS 7의 변화를 주도한 핵심이라고도 할 수 있습니다. 현실 세계를 반영하는 디자인을 추구한다는 스쿼어모피즘(Skeuomorphism)에서 사용자가 원하는 내용과 기능에 집중한다는 미니멀리즘(Minimalism)을 추구하는 플랫폼 디자인으로 바뀌었죠. 이는 iOS 7이 공개된 이후부터 지금까지 많은 이야기를 들었을 것으로 생각하고 자세한 설명은 하지 않겠습니다. 그런데 이렇게 바뀐 디자인에는 개발 환경 면에서도 중요한 이유가 있습니다.

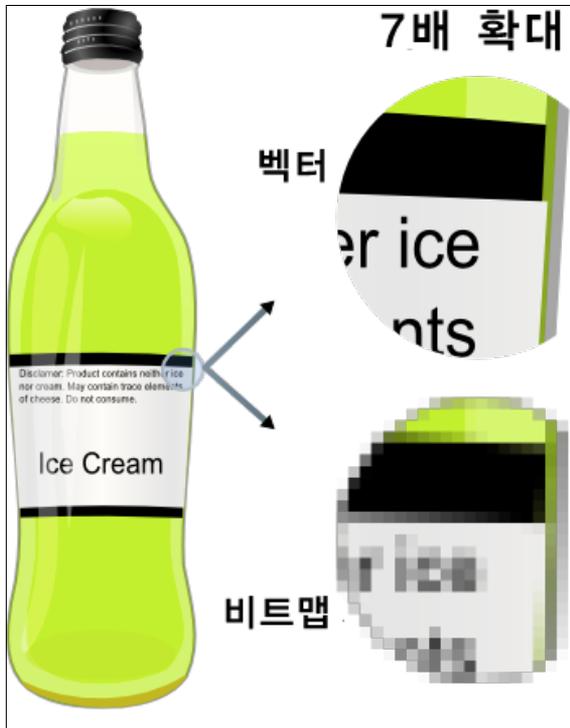
기존 스쿼어모피즘 중심의 디자인을 구현하려면 아무래도 그래픽 효과를 구현할 때 픽셀이라는 점 단위로 색상과 위치 정보를 담는 비트맵 방식을 사용하게 됩니다. 이 방식은 말 그대로 픽셀 단위마다 정보를 담기 때문에 해상도가 다르면 이 정보를 그대로 사용할 수 없습니다. 따라서 다양한 크기의 해상도 비율에 대응하려면 여러 개의 비트맵 이미지를 미리 만들고 이를 대체하는 방식을 사용해야 합니다. 이는 패키징 파일의 크기가 커지고 그에 따른 정보 처리의 양이 늘어나므로 시스템 리소스를 많이 점유하는 비효율적인 상황이 발생하기 때문이죠. iOS도 이제 안드로이드와 같은 해상도 비율의 파편화를 피할 수 없게 되었고 앞으로 발표될 예상 라인업을 생각해보면 비효율적인 상황이 발생하는 것은 자명합니다.

그림 1-2 비트맵 방식의 그래픽 효과 구현(출처: 위키피디아)



이러한 상황에서 플랫 UI는 훌륭한 해결책이 됩니다. 플랫 UI는 비트맵 대신 벡터 방식의 그래픽 효과를 구현합니다. 선과 면의 좌표 값 정보로 전체상을 구성해서 그래픽 효과를 구현하기 때문에 이미지를 확대하거나 축소할 때 품질의 변화가 없고 값 정보만 저장해 저장 용량이 작다는 장점이 있습니다. 이는 다양한 크기의 해상도 비율이 바뀌더라도 값 정보만 저장되어 있다면 비트맵 방식보다 훨씬 빠르고 효율적으로 다양한 크기의 해상도 비율에 대응할 수 있다는 의미입니다. 이번 iOS 7의 플랫 UI와 함께 디자인 구현 시 매우 강력한 권장 사항인 자동 레이아웃은 이 벡터 그래픽 방식을 차용했습니다.

그림 1-3 벡터 방식의 그래픽 효과 구현(출처: 위키피디아)



플랫 UI는 이처럼 단순한 디자인의 변화는 물론이고 장차 다양한 iOS 기기의 그래픽 효과 구현에 있어 개발상의 효율을 추구했다는 사실을 꼭 기억하기 바랍니다.

1.3 64비트 지원

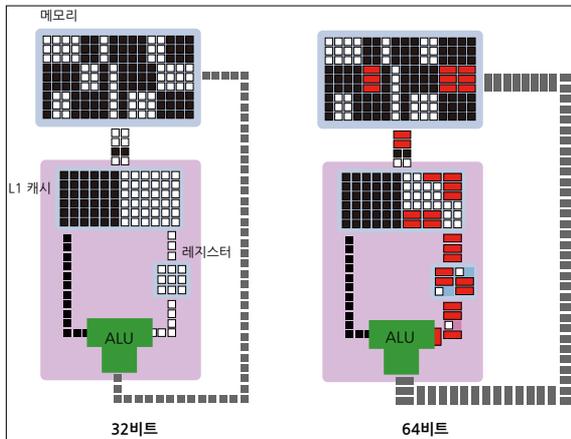
iOS 7, 정확히 말해 iOS 7과 A7 칩을 탑재한 모든 iOS 기기는 이제 64비트를 지원합니다. A7 칩의 기반인 ARMv8 아키텍처는 레지스터가 두 배가 되었고, 이로 인해 한 번에 처리할 수 있는 연산의 양이 늘어났습니다.

단, 아직까지 32비트 기반의 기기와 앱이 대다수인 현실을 고려하면 아직 64비트 앱이 보편화되는 것은 시기상조가 아닐까라는 생각도 듭니다. 애플은 기존 64비트 앱으로의 변환 방법을 담은 가이드를 배포하며, 당분간 A7 칩을 탑재하지 않은 기기에는 유니버설 바이너리 형태로 64비트와 32비트를 동시에 지원한다고 합니다.

이런 경우 64비트 앱의 성능을 제대로 발휘할 수 있는 기기는 A7 칩을 탑재한 일부 기기뿐이며, 이는 32비트 기기에서는 64비트 앱의 성능을 제대로 누리지 못한다는 것이 문제입니다. 앞으로 A7 혹은 그 이상의 상위 칩을 탑재한 iOS 기기가 늘어나면서 점차 자연스럽게 해결되기는 하겠지만 이 책을 쓴 2013년 말에서 2014년도까지는 아직 과도기 단계라고 보는 것이 정확할 것입니다.

그리고 64비트와 32비트의 개념을 정확히 이해해야 합니다. 64비트와 32비트의 가장 큰 차이는 한 번에 처리할 수 있는 데이터의 전송량과 이를 처리하는 단위가 커졌다는 것입니다. 즉, 처리할 데이터의 양이 작은 앱이라면 사실 32비트나 64비트나 아주 큰 성능상의 이점을 누리는 것은 아니라는 뜻입니다.

그림 1-4 32비트와 64비트의 차이(출처: arstechnica.com)



64비트로 큰 이득을 보는 분야는 동영상이나 오디오 인코딩, 그리고 그래픽 효과에 크게 의존하는 일부 게임 등입니다. 애플이 밝혔듯이 사진 앨범의 연도별 타일 처리나 동영상 처리, 그리고 WWDC 2013에 iOS 7 발표와 함께 공개한 인피니티 블레이드 Infinite Blade 3 등의 성능 향상에도 큰 도움을 줍니다.

애플이 64비트를 지원하는 큰 이유는 사실 iOS 7의 방향성이 앞으로 데스크톱 못지않은, 그리고 OS X와도 비견될 수 있는 다양한 앱을 개발하는 토대를 구축하는데 있다고 생각합니다. 즉, 당장은 64비트가 큰 도움이 되지는 않더라도 미래까지 생각하면 이제 모바일 환경에서도 64비트에 기반을 둔 앱 개발은 필수가 될 것으로 전망할 수 있습니다.

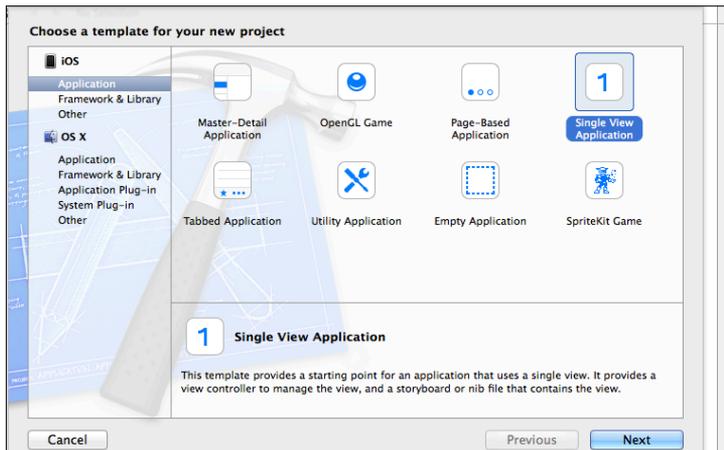
2 | Xcode 5

iOS 7이 발표되면서 많은 iOS 개발자들이 큰 정신적 충격과 패닉에 빠졌습니다. 2장에서는 iOS 7의 개발을 책임질 통합 개발 환경^{IDE}인 Xcode의 달라진 점들을 간략히 살펴보려 합니다(이 책에서 다루는 Xcode의 버전은 5.0.2를 기준으로 합니다).

2.1 새롭게 추가된 템플릿

Xcode 4와 비교하면 기존의 템플릿은 그대로 유지된 상태에서 SpriteKit Game이라는 새로운 템플릿이 추가되었습니다. iOS 7에서 새롭게 추가된 SpriteKit 프레임워크에 대응하는 템플릿입니다. 자세한 설명은 ‘4.7 스프라이트킷 프레임워크’를 참고하기 바랍니다.

그림 2-1 iOS 7의 주요 프로젝트 템플릿



2.2 새로워진 문서먼트

Xcode에서 가장 많이 사용하는 기능을 꼽으라면 저는 주저 없이 문서먼트를 이야기합니다. 개발하다가 모르는 iOS의 라이브러리를 찾아보려면 당연히 문서먼트를 열기 때문입니다. 보통 능숙한 개발자라면 단축키를 사용해서 문서먼트를 열겠지만, iOS 개발을 처음 접하는 분이라면 주로 Organizer를 실행해 문서먼트를 찾아볼 것입니다. 그런데 문서먼트가 항상 있던 Organizer에서 사라졌습니다. 즉, ‘Documentation’ 탭이 사라졌습니다.

그림 2-2 Xcode 4의 Organizer

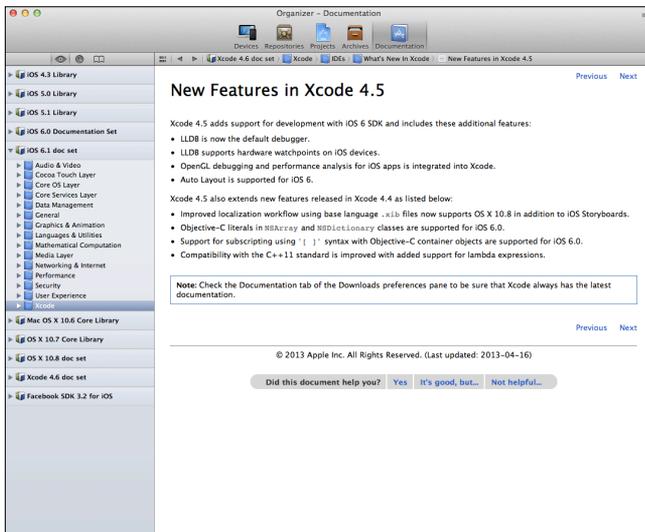
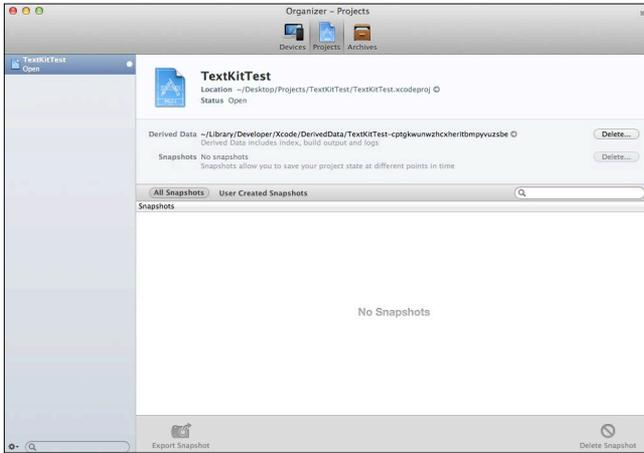


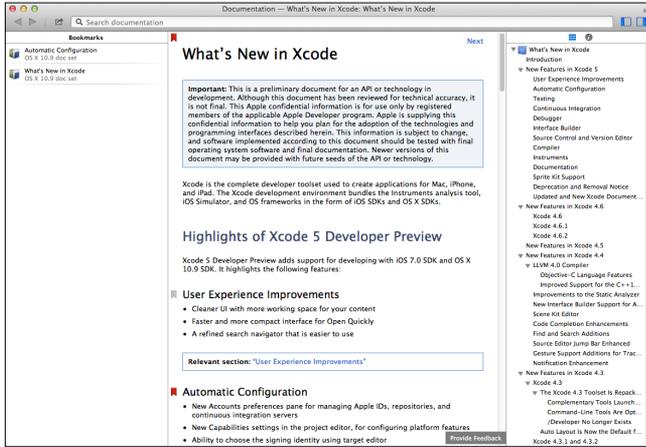
그림 2-3 Xcode 5의 Organizer



하지만 도큐먼트는 기존처럼 [Help] 메뉴 혹은 단축키 [alt + command + ?]로 도큐먼트를 실행할 수 있습니다. 처음 실행하면 'New Features in Xcode 5' 도큐먼트를 확인할 수 있습니다.

또한 새로운 창으로 변모함과 동시에 책갈피도 바뀌었습니다. 기존의 책갈피는 문서별로 지정해줄 수 있었지만, 이제는 특정 섹션별로 책갈피를 따로 설정해줄 수 있습니다. 그림 2-4처럼 세션이 파란색의 책갈피로 지정된 것을 볼 수 있습니다.

그림 2-4 새로운 문서먼트 창

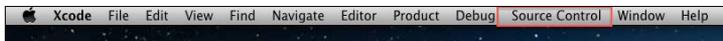


2.3 Source Control

최근에는 소스 코드의 버전 관리를 위해 Git 또는 subversion을 사용하는 프로젝트가 많습니다. 필자 역시 실무에서 버전 관리 시스템을 사용하는데, 지금까지는 이를 Organizer의 'Repositories' 탭에서 관리했습니다. 그런데 그림 2-3에서도 확인했듯이 Organizer에서는 'Repositories' 탭에서도 없어졌습니다.

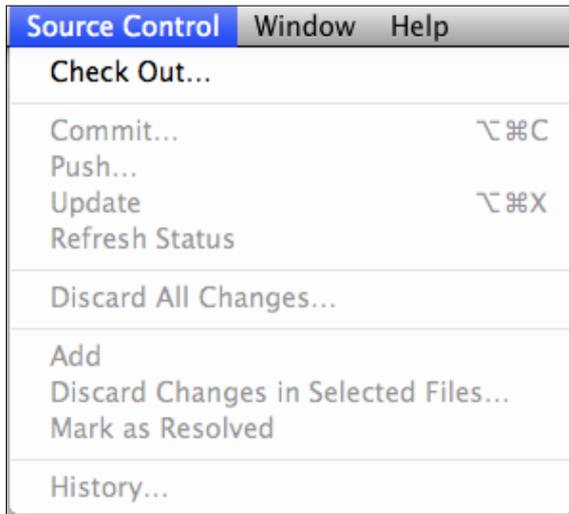
대신 [Source Control]이라는 메뉴가 생겼습니다.

그림 2-5 [Source Control] 메뉴



그런데 그림 2-6에서 볼 수 있듯이 [Source Control]을 눌렀을 때 동작하는 기능은 [Check Out...] 뿐이고 [Commit], [Push], [Update] 등의 메뉴는 비활성화되어 있습니다.

그림 2-6 비활성화된 [Source Control] 메뉴

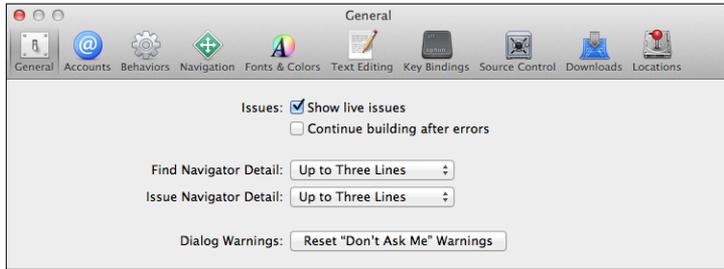


비활성화된 이유는 '2.4 편리해진 계정 관리'에서 살펴보겠습니다.

2.4 편리해진 계정 관리

앞에서 사용한 [Source Control]의 메뉴가 활성화되려면 'Preference'를 살펴 봐야 합니다. 'Preference' 역시나 약간의 변화가 있습니다. 'Accounts'와 'Navigation' 탭이 추가되었습니다. 계정 관리를 위해 살펴볼 탭은 'Accounts' 탭입니다.

그림 2-7 Xcode 5의 Preference



'Accounts' 탭을 선택한 후 [+] 버튼을 누르고 'Add Repository'를 선택하면 Repository 계정 정보를 입력할 수 있습니다. Git 혹은 Subversion이라는 Type 을 선택한 다음 원하는 프로젝트를 연결할 수 있습니다.

그림 2-8 새로운 Repository 계정 설정

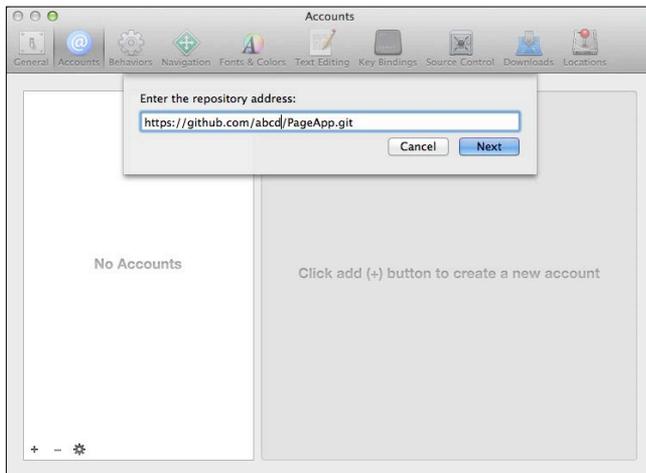
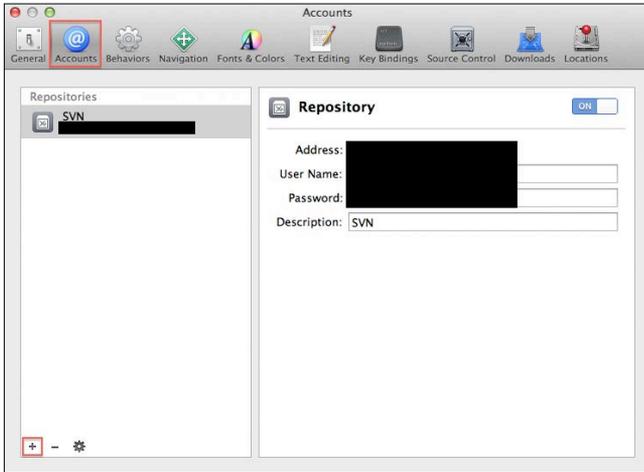


그림 2-9 Repository 계정 설정 확인



그뿐만 아니라 Git이나 Subversion 이외에 애플 아이디도 추가할 수 있게 되어 있습니다. 별도의 소스 코드 관리 서버 정보도 추가해 줄 수 있습니다.

그림 2-10 다양한 계정 등록



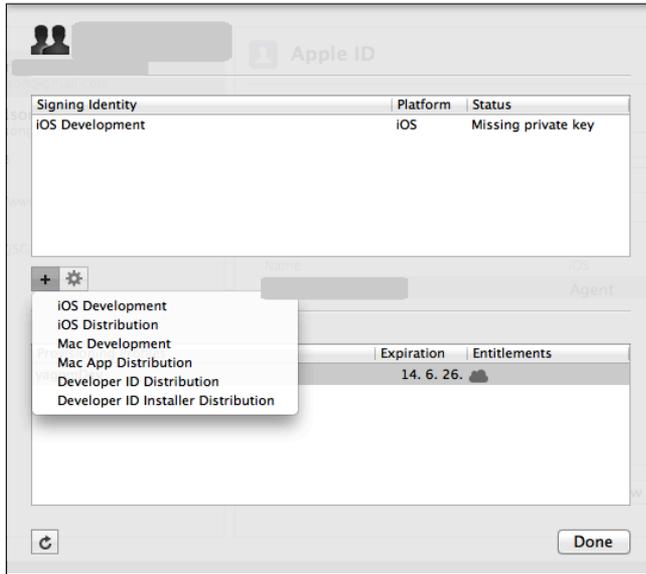
애플 아이디는 애플 개발자 프로그램에 가입한 애플 아이디를 추가하면 됩니다. [View Details]를 누르면 iOS와 Mac 관련 개발자 서명이나 개발/배포 프로비저닝 프로파일을 관리할 수 있습니다.

특히 처음 Xcode를 설치한 컴퓨터에는 iOS 개발 센터에 등록된 개발자 서명 정보나 프로비저닝 프로파일 등을 다운로드할 수도 있습니다. iOS 개발 센터에 들어갈 일이 하나 줄게 되는 셈입니다.

그림 2-11 개발자 서명과 프로비저닝 프로파일 정보 확인



그림 2-12 개발/배포 서명서 정보 확인



이제 [Source Control] → [Check Out]을 누른 후 'Accounts' 탭에서 등록된 Repositories을 선택할 수 있습니다. 'Repositories'을 선택한 다음 [Next]를 누르면 컴퓨터의 원하는 폴더에 프로젝트를 저장할 수도 있습니다.

Xcode는 갈수록 크나큰 발전을 거듭하는 것이 아닌가 싶습니다.

그림 2-13 프로젝트 등록

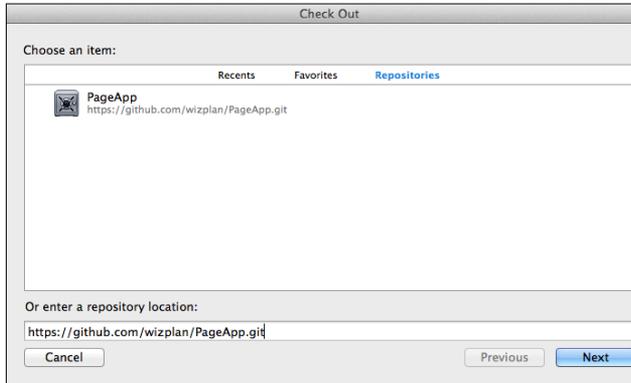


그림 2-14 실행된 프로젝트와 활성화된 [Source Control] 메뉴 항목

